Variables and Arithmetic

Remix CS 2019-20

What we will cover today

- What are variables and how to use them
- Basic arithmetic in Python
- The modulus operator and integer division

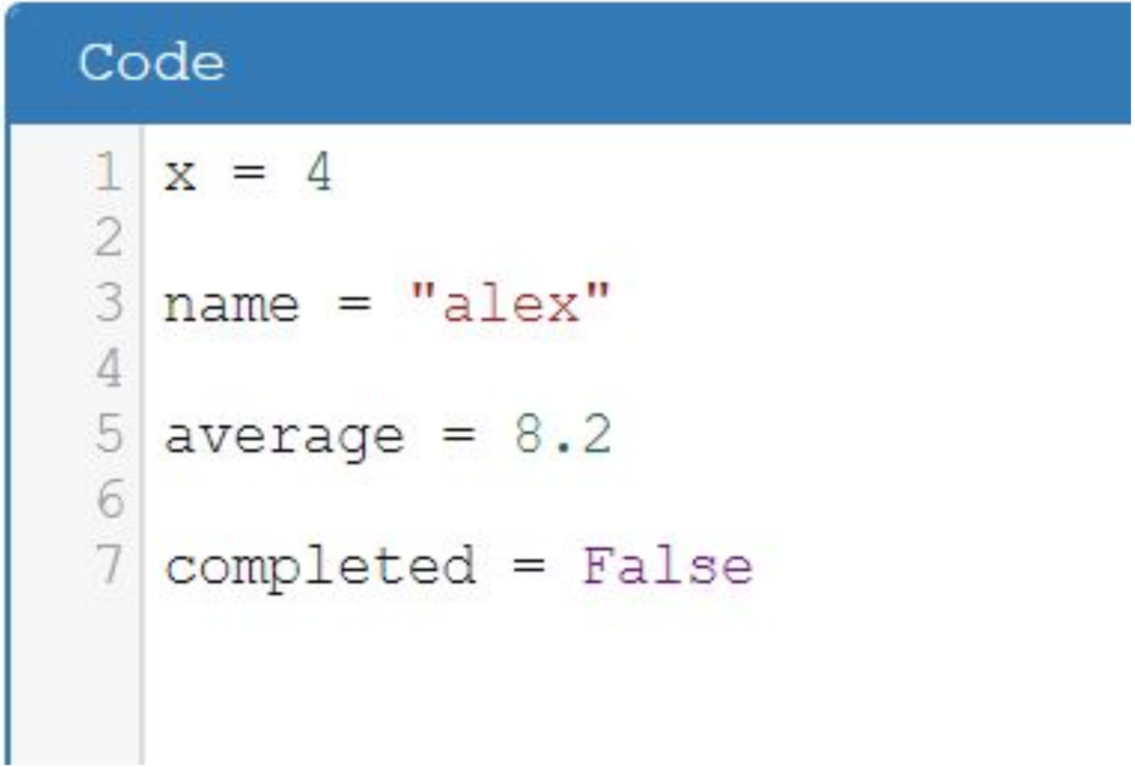
What are variables?

- Variables are used to **store data** as a value that code can refer back to any time an instruction needs to read it or change it
- Why do we use them?
  - avoids repeating value in an instruction; we can reuse the variable instead **(reusable)**
  - we can give data a name that is clear for a coder to understand **(human-readable)**

How you use variables in Python

- In Python, variables are assigned using the assignment operator
- assignment operator: the ‘equals sign’ =
- Example: **x = 4**
    - variable name → x
    - variable data → 4
    - variable type → int
- Example: **name = “alex”**
    - variable name → name
    - variable data → “alex”
    - variable type → str

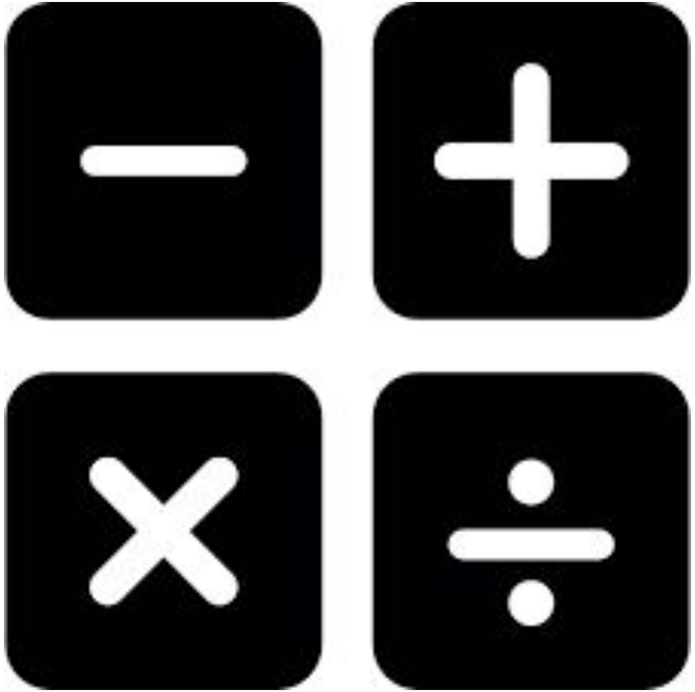
What this looks like in Python



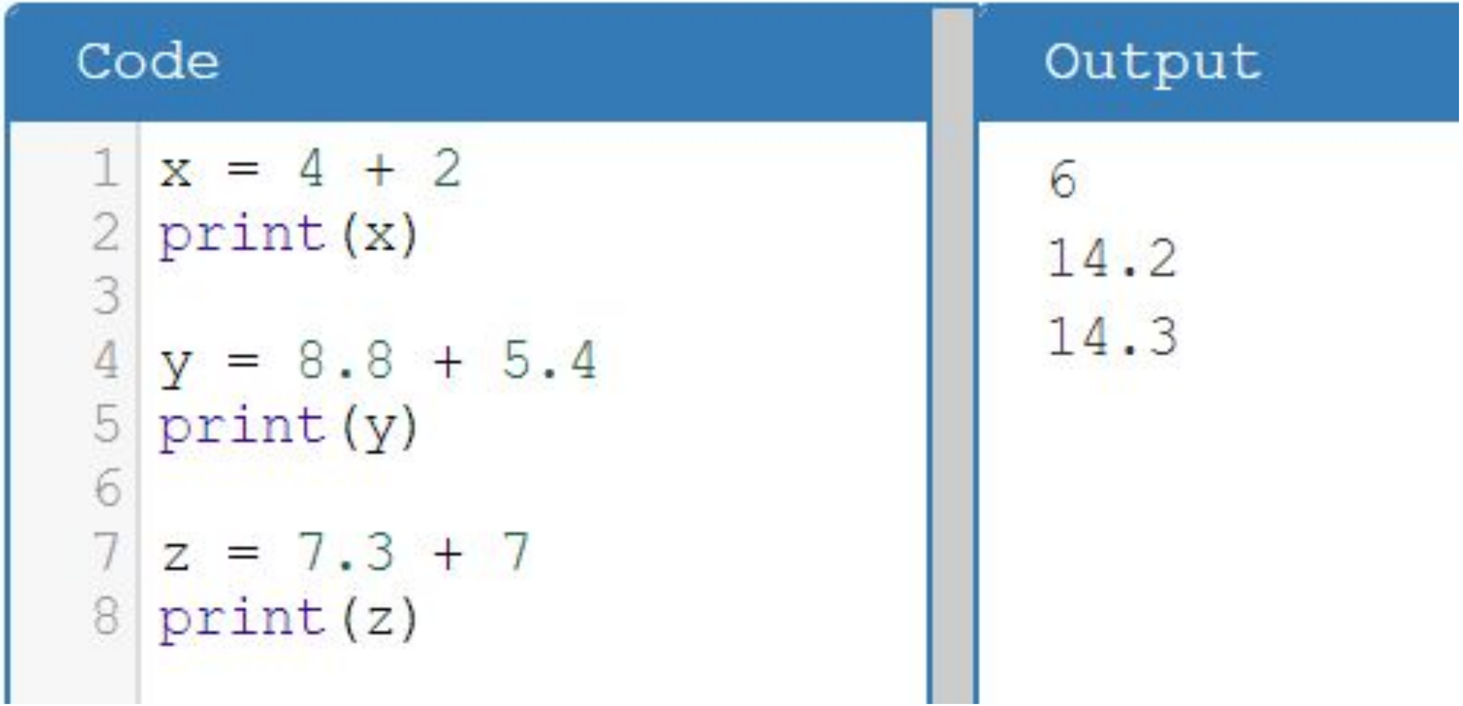
```python
x = 4

name = "alex"

average = 8.2

completed = False
```

Basic Arithmetic in Python

- Addition
- Subtraction
- Multiplication
- Division
- Exponentiation



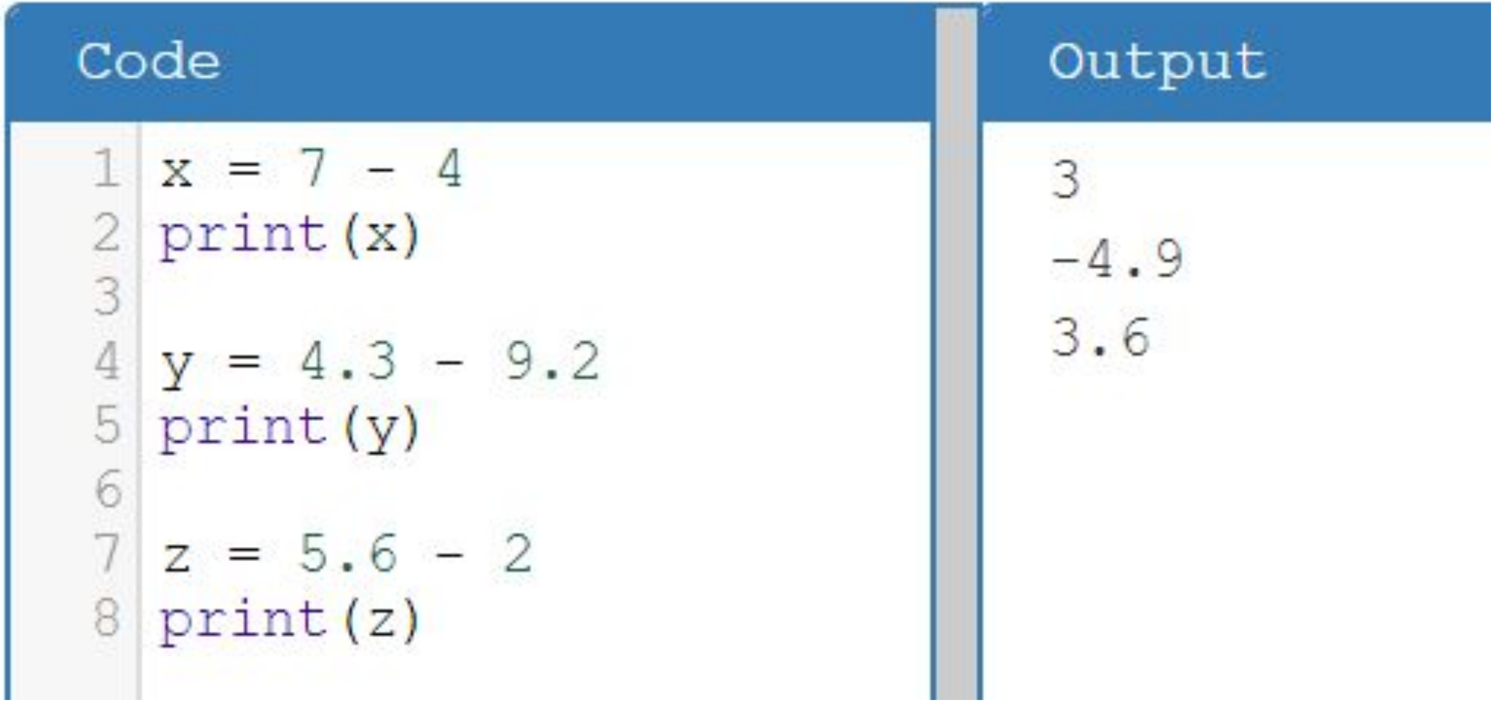
Addition: use + sign



```
x = 4 + 2
print(x)

y = 8.8 + 5.4
print(y)

z = 7.3 + 7
print(z)
```

```
6
14.2
14.3
```

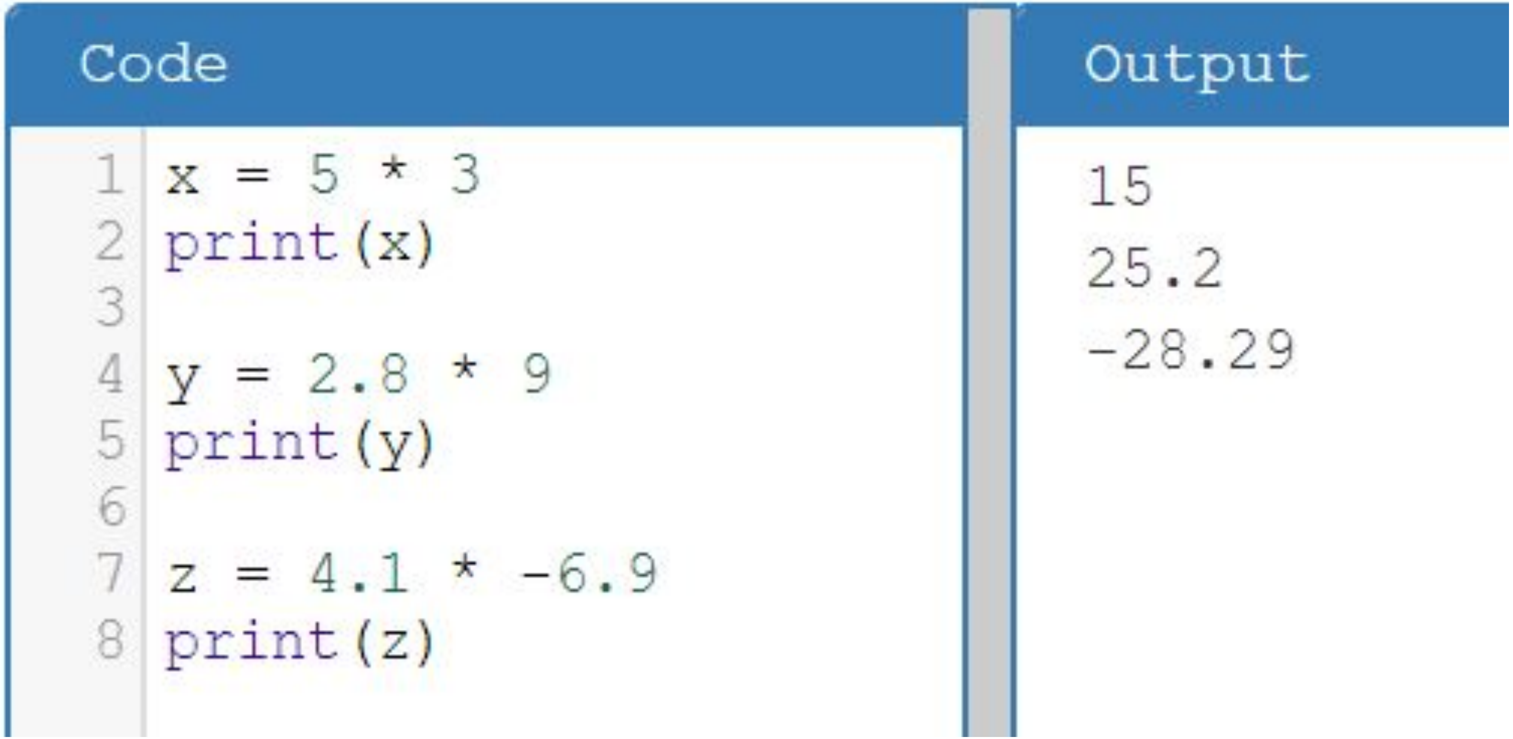
Subtraction: use - sign



```python
x = 7 - 4
print(x)

y = 4.3 - 9.2
print(y)

z = 5.6 - 2
print(z)
```

```
3
-4.9
3.6
```

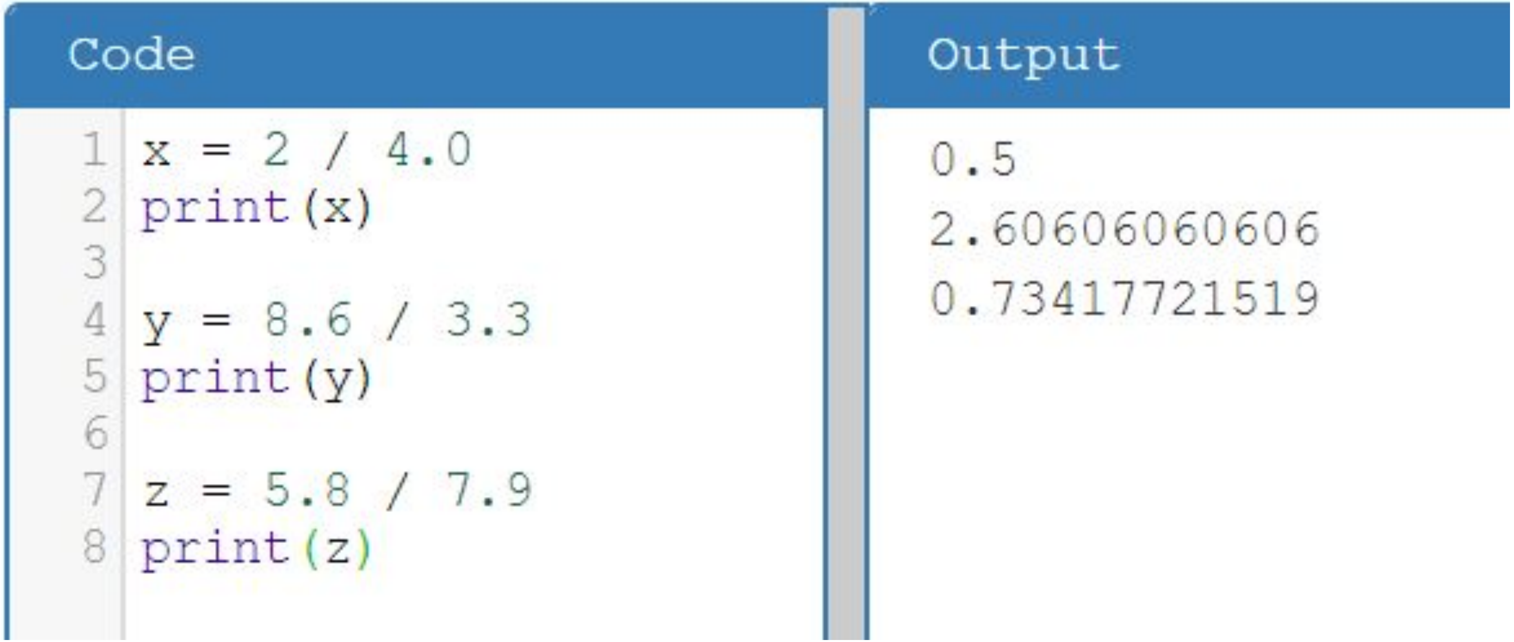
Multiplication: use *



```
x = 5 * 3
print(x)

y = 2.8 * 9
print(y)

z = 4.1 * -6.9
print(z)
```

```
15
25.2
-28.29
```

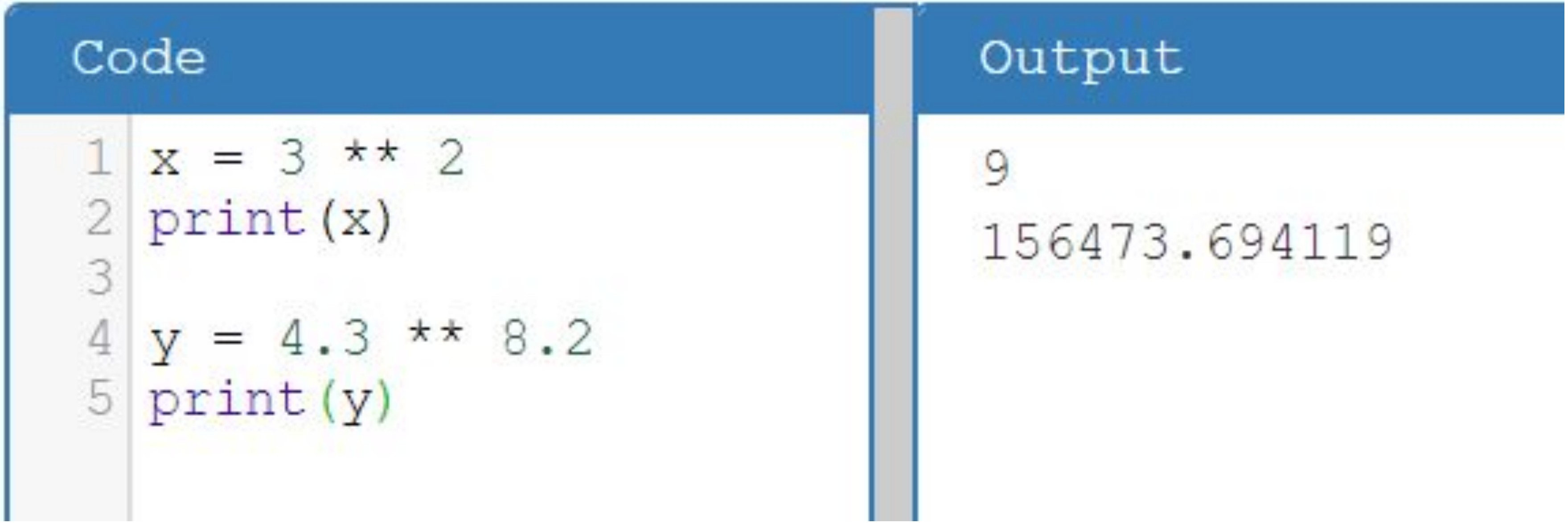
Division: use /



```python
x = 2 / 4.0
print(x)

y = 8.6 / 3.3
print(y)

z = 5.8 / 7.9
print(z)
```

```
0.5
2.60606060606
0.73417721519
```

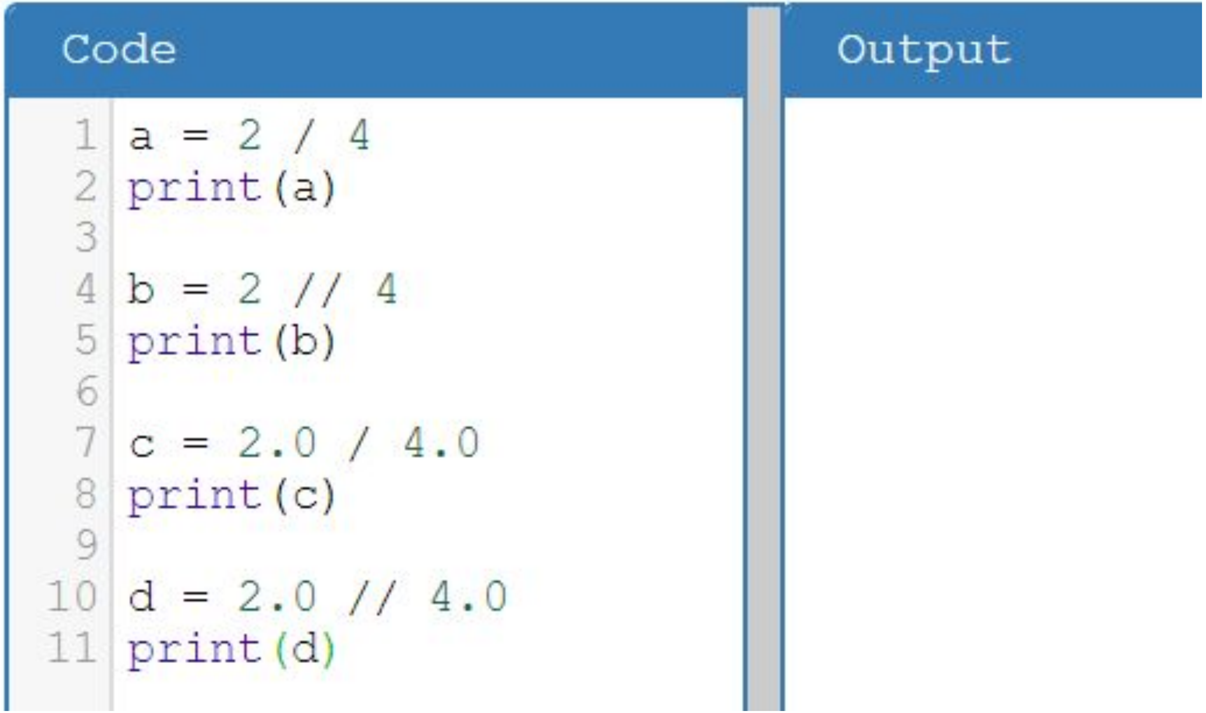
Exponents: use **



```
x = 3 ** 2
print(x)

y = 4.3 ** 8.2
print(y)
```

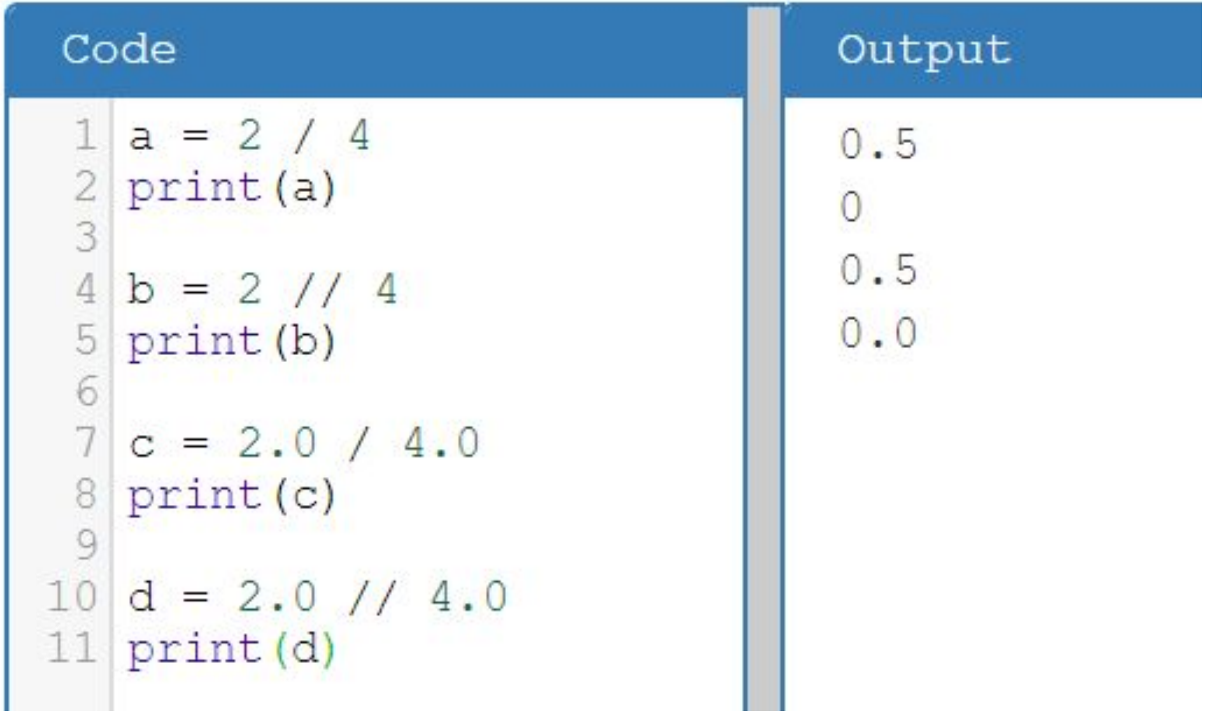
```
9
156473.694119
```

Integer Division: use // - try the following



```python
a = 2 / 4
print(a)

b = 2 // 4
print(b)

c = 2.0 / 4.0
print(c)

d = 2.0 // 4.0
print(d)
```

Integer Division: here’s what you should see



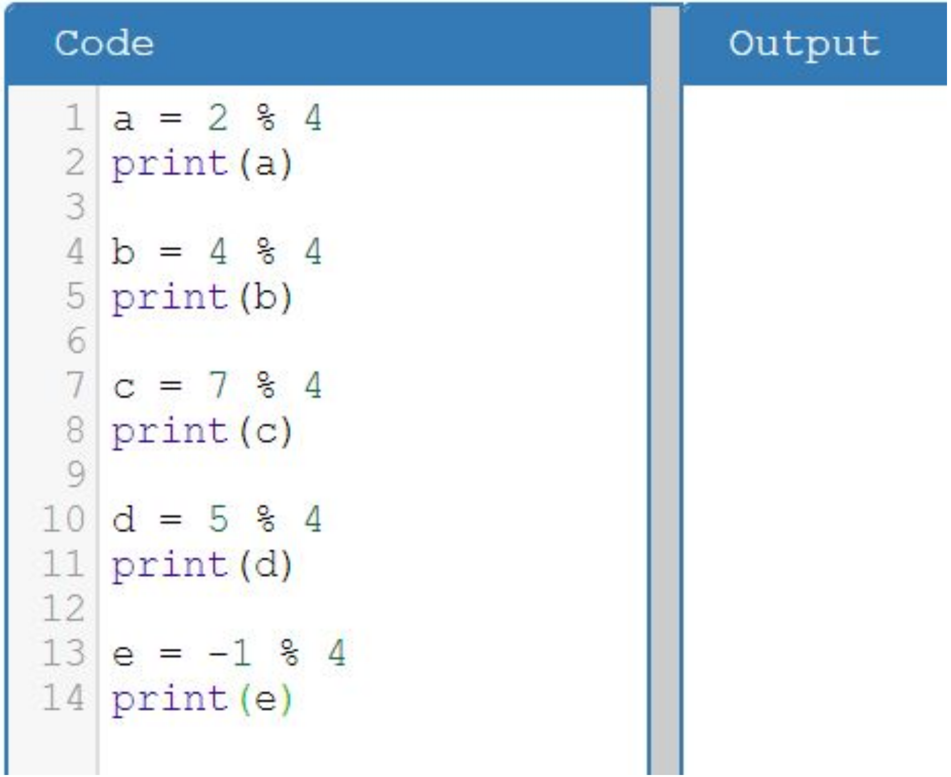
```python
a = 2 / 4
print(a)

b = 2 // 4
print(b)

c = 2.0 / 4.0
print(c)

d = 2.0 // 4.0
print(d)
```

```
0.5
0
0.5
0.0
```

So what is Integer Division?

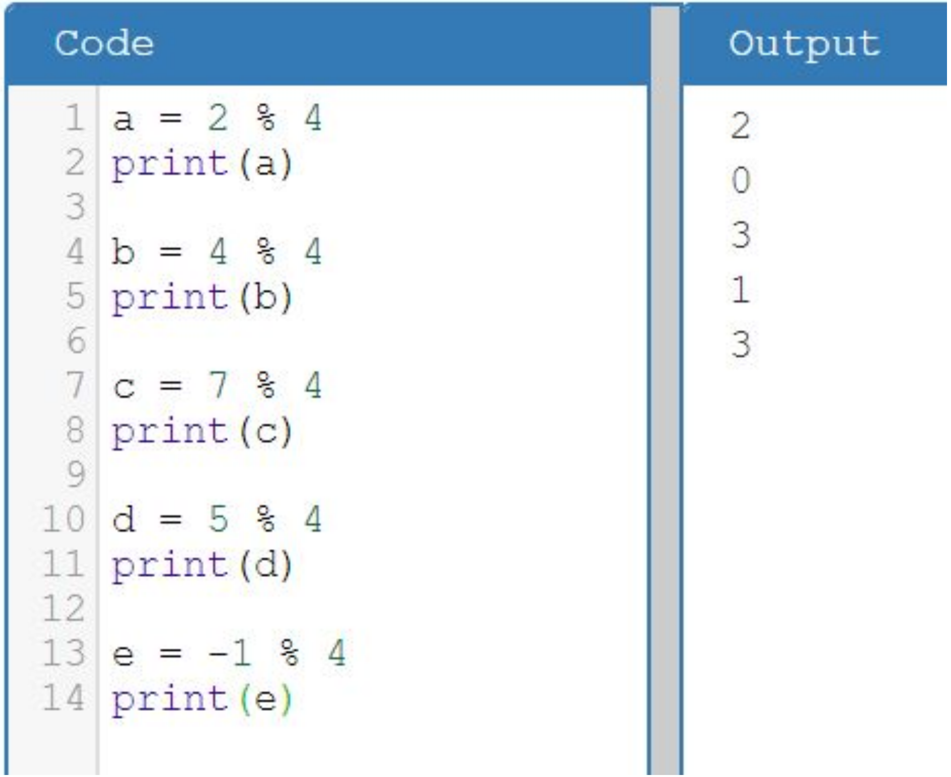
- Integer division means you are finding the greatest **whole** number of times one number divides into another
- So back to the example, 4 goes into 2 half a time, which is not a whole number, to the whole number (integer) times that 4 goes into 2 is 0
- Similarly 9 // 4 would evaluate to 2 not 2.25

Modulus Operator: use % - try the following



```
a = 2 % 4
print(a)

b = 4 % 4
print(b)

c = 7 % 4
print(c)

d = 5 % 4
print(d)

e = -1 % 4
print(e)
```

Modulus Operator: here’s what you should see



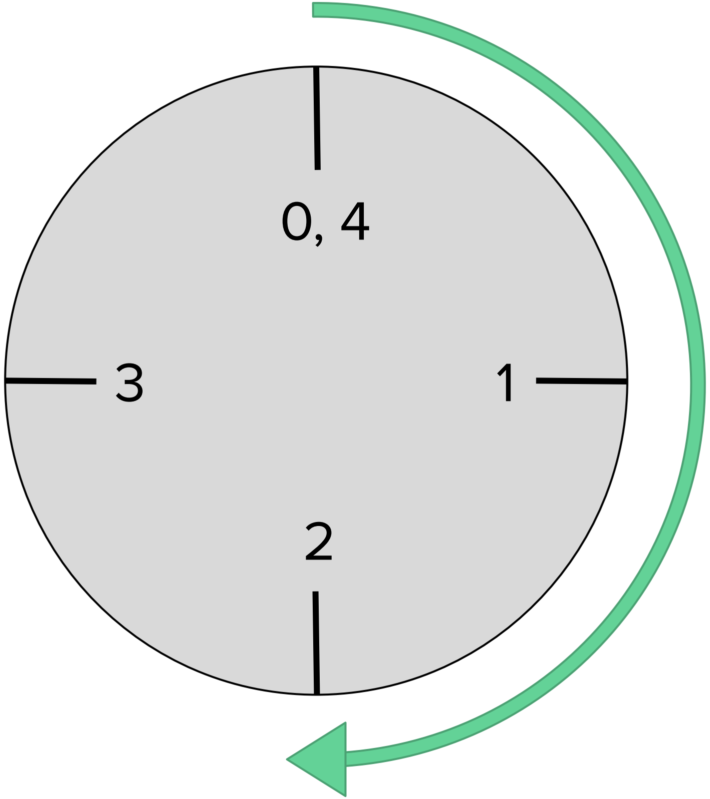
```
a = 2 % 4
print(a)

b = 4 % 4
print(b)

c = 7 % 4
print(c)

d = 5 % 4
print(d)

e = -1 % 4
print(e)
```

```
2
0
3
1
3
```

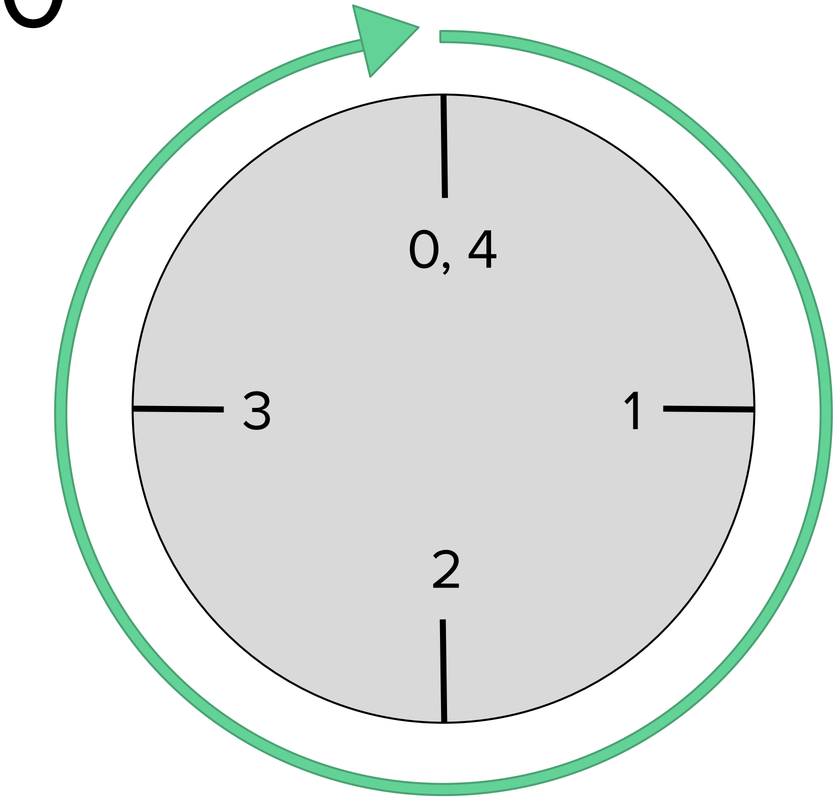
So what is the Modulus Operator?

- There are 2 main ways to think about it
- One is that it is like the remainder when you divide the first number by the second number
- The other way is like a clock (see subsequent slides) where the second number is the number of hours, and the time wraps around once you exceed it (for example there is no 27 o’clock)

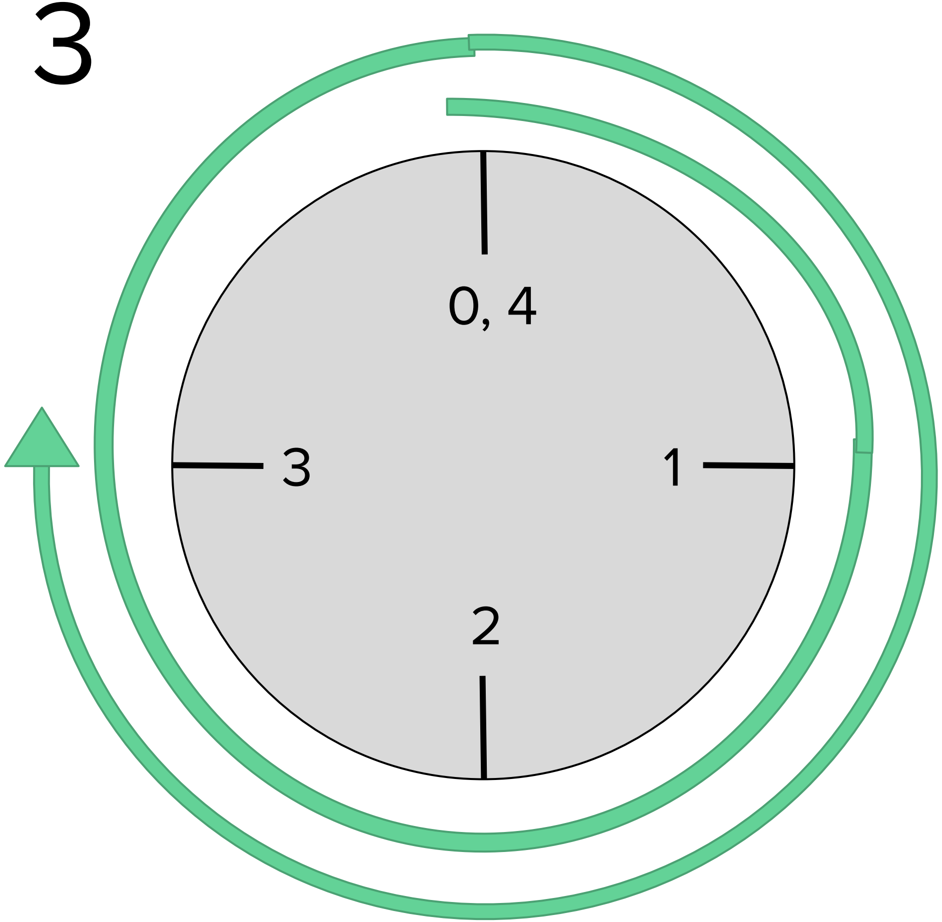
2 % 4 = 2



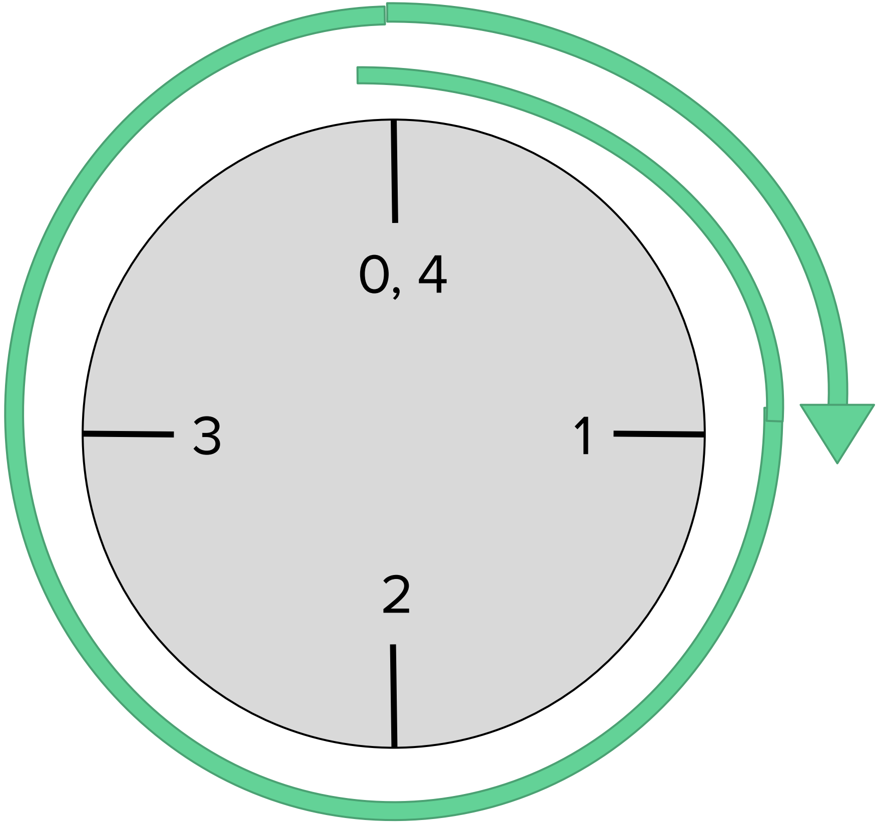
4 % 4 = 0



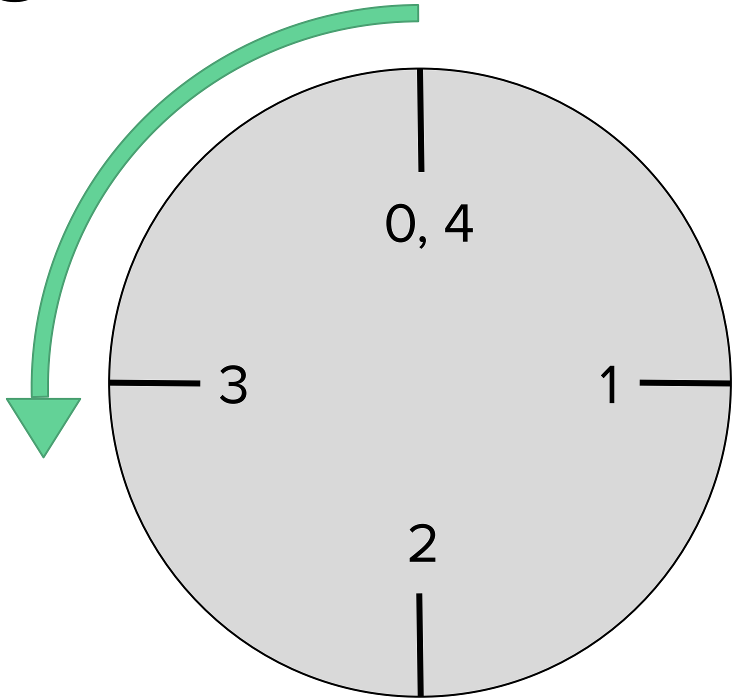
7 % 4 = 3



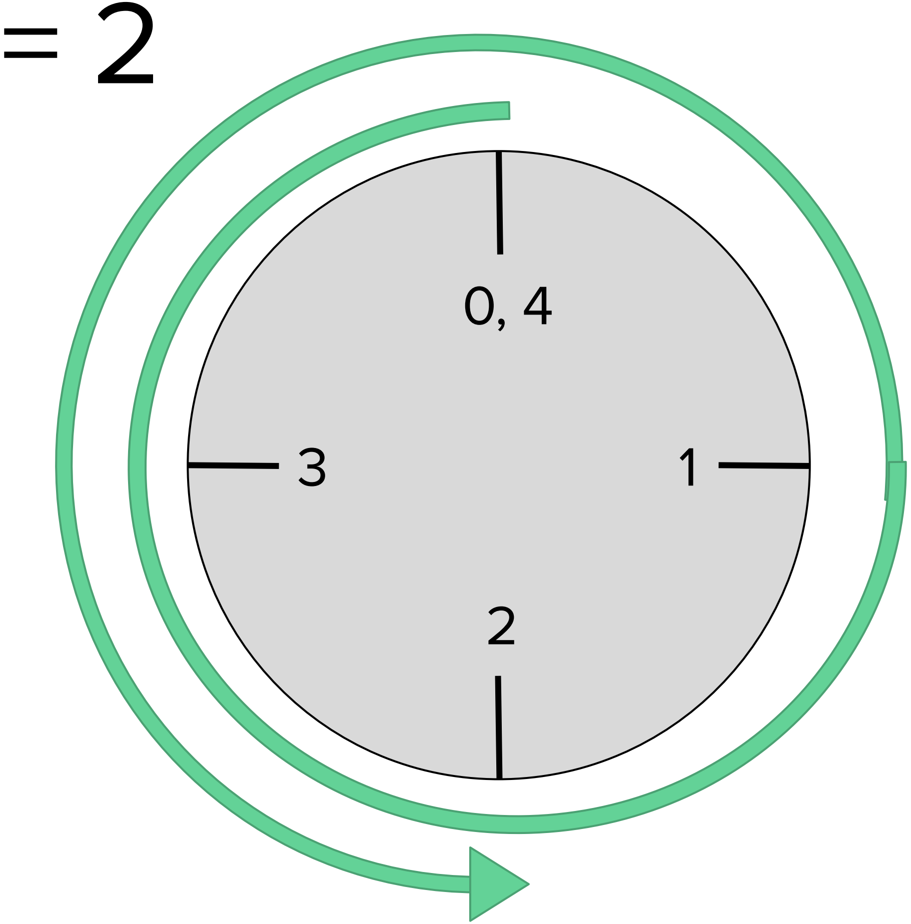
5 % 4 = 1



$-1 \% 4 = 3$

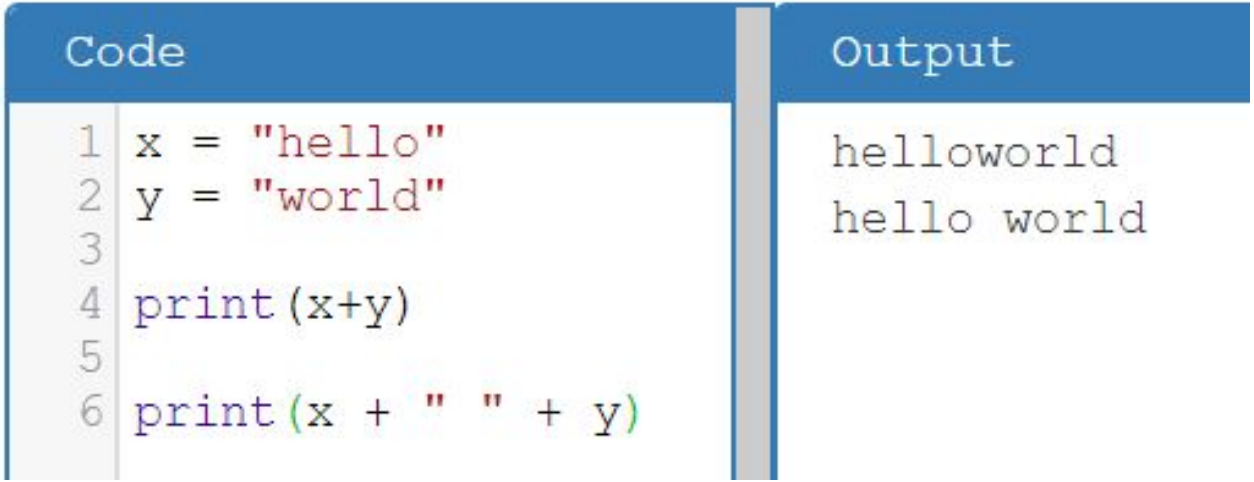


-6 % 4 = 2



String Concatenation

- You can join multiple strings together with the + operator

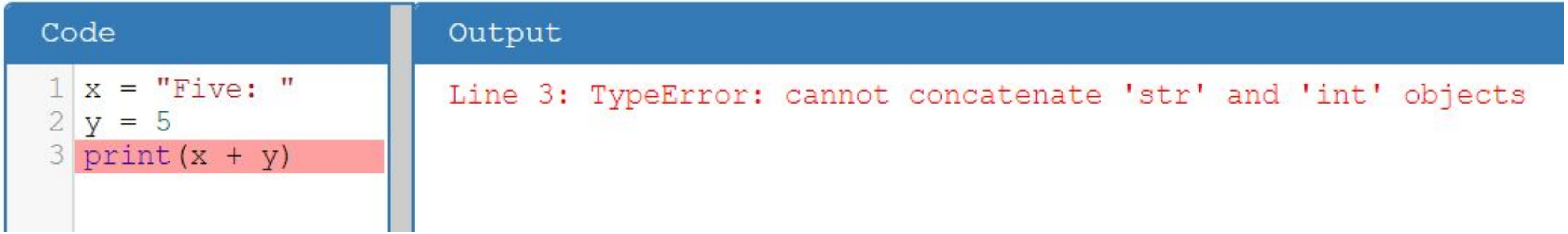


```python
x = "hello"
y = "world"

print(x+y)

print(x + " " + y)
```

```
helloworld
hello world
```

String Concatenation with Non Strings

- What happens if you try to join a string and an integer?

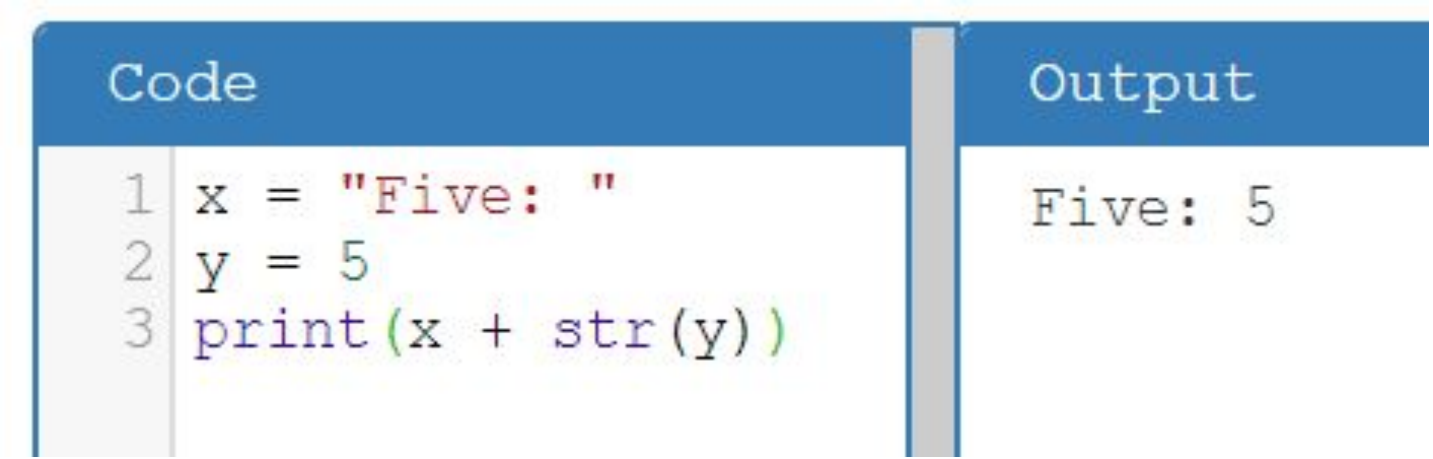


```
1 x = "Five: "
2 y = 5
3 print(x + y)
```

```
Line 3: TypeError: cannot concatenate 'str' and 'int' objects
```

String Concatenation with Non Strings

- The solution: using the str() function to transform the integer into a string

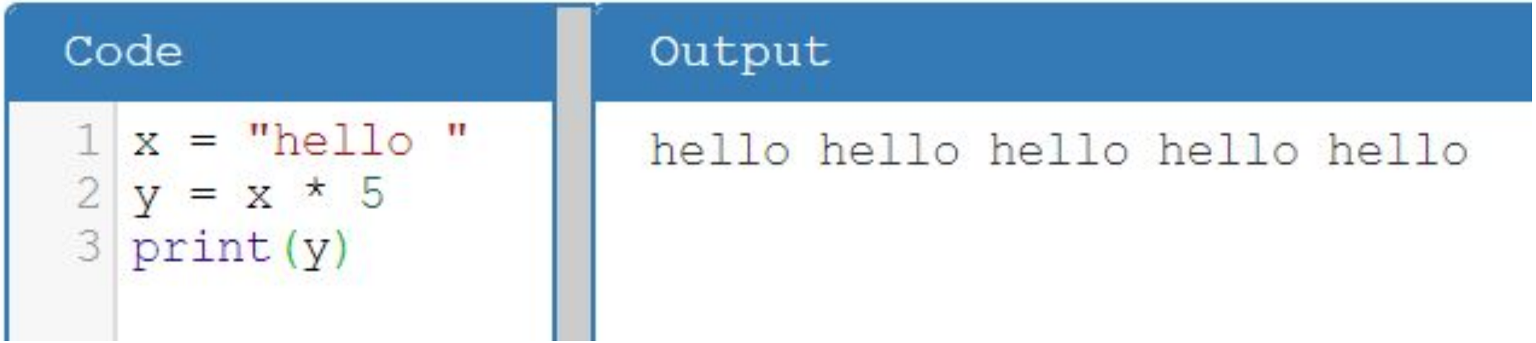


```python
x = "Five: "
y = 5
print(x + str(y))
```

```
Five: 5
```

Repeating Strings

- One more fun trick you can do with strings
- The multiplication operator repeats a given string the specified number of times



```
x = "hello "
y = x * 5
print(y)
```

```
hello hello hello hello hello
```

Practice Exercise

- Given a 3 digit positive integer, write a program that will print out the value of the hundreds place, the tens place, and the ones place
    - Ex: **x** = **328** should print **“Hundreds place is 3, Tens place is 2, Ones place is 8”**
- Your output should be one line, so you will have to use string concatenation as well as the modulus and integer division operators.

Practice Exercise

- Given a number of seconds since the start of the day, write a program that will print out the current time in a 12 hour time format.
  - Ex: **seconds** = **13782** should print out **3:49:42**
  - Ex: **seconds** = **68682** should print out **7:4:42**
  - Note that you can ignore the AM or PM for this exercise, for instance the second example above is equivalent to 7:4:42 PM, whereas the first example is 3:49:42 AM